The Hamilton-Jacobi Skeleton

Kaleem Siddiqi* Sylvain Bouix† Allen Tannenbaum‡ Steven W. Zucker§

Abstract

The eikonal equation and variants of it are of significant interest for problems in computer vision and image processing. It is the basis for continuous versions of mathematical morphology, stereo, shape-from-shading and for recent dynamic theories of shape. Its numerical simulation can be delicate, owing to the formation of singularities in the evolving front, and is typically based on level set methods. However, there are more classical approaches rooted in Hamiltonian physics, which have received little consideration in computer vision. In this paper we first introduce a new algorithm for simulating the eikonal equation, which offers a number of computational and conceptual advantages over the earlier methods when it comes to shock tracking. Next, we introduce a very efficient algorithm for shock detection, where the key idea is to measure the net outward flux of a vector field per unit volume, and to detect locations where a conservation of energy principle is violated. We illustrate the approach with several numerical examples including skeletons of complex 2D and 3D shapes.

1. Introduction

Variational principles emerged naturally from considerations of energy minimization in mechanics [11]. We consider these in the context of the eikonal equation, which arises in geometrical optics and, recently, which has become of great interest for problems in computer vision [4]. It is the basis for continuous versions of mathematical morphology [3, 18, 25], as well as for Blum's grassfire transform [2] and new dynamic theories of shape representation including [9, 23]. It has also been widely used for applications in image processing and analysis [19, 5], shape-from-shading [10] and stereo [8].

The numerical simulation of this equation is non-trivial, because it is a hyperbolic partial differential equation for which a smooth initial front may develop singularities or *shocks* as it propagates. At such points, classical concepts such as the normal to a curve, and its curvature, are not defined. Nevertheless, it is precisely these points that are important for the above applications in computer vision since, e.g., it is they which denote the skeleton (see Figure 3). To continue the evolution while preserving shocks, the technology of level set methods introduced by Osher and Sethian [15], has proved to be extremely powerful. The approach relies on the notion of a weak solution, developed in viscosity theory [6], and the introduction of an appropriate entropy condition to select it. The representation of the evolving front as a level set of a hypersurface allows topological changes to be handled in a natural way, and robust, efficient implementations have recently been developed [20].

Level set methods are Eulerian in nature because computations are restricted to grid points whose locations are fixed. For such methods, the question of computing the locus of shocks for dynamically changing systems remains of crucial importance, i.e., the methods are shock *preserving* but do not explicitly *detect* shocks. Shock detection methods which rely on interpolation of the underlying hypersurface are computationally very expensive. Numerical thresholds are introduced and high order accurate numerical schemes must be used [14, 22].

On the other hand, there are more classical methods rooted in Hamiltonian physics, which can also be used to study shock theory. To the best of our knowledge, these have not been considered in the computer vision literature. The purpose of this paper is twofold. First, we introduce the above methods and a straightforward algorithm for simulating the eikonal equation, which offers a number of computational and conceptual advantages when it comes to shock tracking. The proposed algorithm is Lagrangian in nature, i.e., the front is explicitly represented as a sequence of marker particles. The motion of these particles is then governed by an underlying Hamiltonian system. Such systems are of course fundamental in classical physics, and have a natural physical interpretation based on elementary Hamiltonian and Lagrangian mechanics. Second, we introduce a very efficient algorithm for shock detection based on the net

*School of Computer Science & Center for Intelligent Machines McGill University, Montréal, PQ, Canada H3A 2A7

†School of Computer Science & Center for Intelligent Machines McGill University, Montréal, PQ, Canada H3A 2A7

‡Department of Electrical and Computer Engineering, University of Minnesota, 200 Union Street S. E. Minneapolis, MN 55455

§Department of Computer Science & Department of Electrical Engineering, Yale University, New Haven, CT 06520-8285

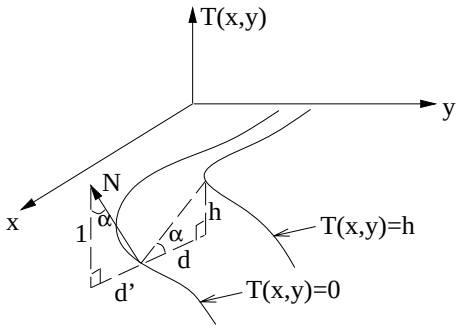

Figure 1: A geometric view of a monotonically advancing front (Eq. 1). $T(x, y)$ is a graph of the 'solution' surface, the level sets of which are the evolved curves.

outward flux per unit volume of the vector field underlying the Hamiltonian system.

2. The Eikonal Equation

We begin by showing the connection between a monotonically advancing front, and the well known eikonal equation. Consider the curve evolution equation

$$\frac{\partial \mathcal{C}}{\partial t} = F\mathcal{N}, \tag{1}$$

where $\mathcal{C}$ is the vector of curve coordinates, $\mathcal{N}$ is the unit inward normal, and $F = F(x, y)$ is the speed of the front at each point in the plane, with $F \geq 0$ (the case $F \leq 0$ is also allowed). Let $T(x, y)$ be a graph of the solution surface, obtained by superimposing all the evolved curves in time (see Figure 1). In other words, $T(x, y)$ is the time at which the curve crosses a point (x, y) in the plane. Referring to the figure, the speed of the front is given by

$$F(x, y) = \frac{d}{h} = \frac{1}{\tan(\alpha)} = \frac{1}{d'} = \frac{1}{\|\nabla T\|}.$$

Hence, $T(x, y)$ satisfies the *eikonal equation*

$$\|\nabla T\|\, F = 1. \tag{2}$$

A number of algorithms have been recently developed to solve a quadratic form of this equation, i.e., $\|\nabla T\|^2 = \frac{1}{F^2}$. These include Sethian's fast marching method [20], which relies on an interpretation of Huygens's principle to efficiently propagate the solution from the initial curve, and Rouy and Tourin's viscosity solutions approach [17]. However, neither of these methods address the issue of shock detection explicitly, and more work has to be done to track shocks.

A different approach, which is related to the solution surface $T(x, y)$ viewed as a graph, has been proposed by Shah *et al* [21, 23]. Here the key idea is to use an edge strength

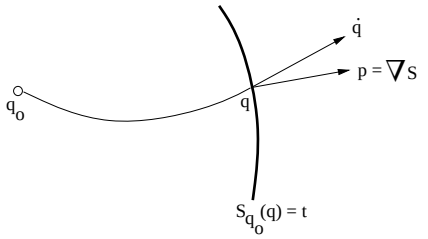

Figure 2: Direction of a ray $\dot{\mathbf{q}}$ and the direction of motion of the wave front $\mathbf{p}$. From [1].

functional v in place of the surface $T(x, y)$, computed by a linear diffusion equation. The equation can be efficiently implemented, and the framework extends to greyscale images as well as curves with triple point junctions. It provides an approximation to the reaction-diffusion space introduced in [9], but does not extend to the extreme cases, i.e., morphological erosion by a disc structuring element (reaction) or motion by curvature (diffusion). Hence, points of maximum (local) curvature are interpreted as skeletal points. This regularized skeleton is typically not connected, and its relation to the classical skeleton, obtained from the eikonal equation with $F = 1$, is as yet unclear.

In the next section, we shall consider an alternate framework for solving the eikonal equation, which is based on the canonical equations of Hamilton. The technique is widely used in classical mechanics, and rests on the use of a Legendre transformation (see [1] for the precise definition) which takes a system of n second-order differential equations to a (mathematically equivalent) system of $2n$ first-order differential equations. We believe that for a number of vision problems involving shock tracking and skeletonization, this represents a natural way of implementing the eikonal equation.

3. Hamilton's Canonical Equations

Following Arnold [1, pp. 248–258], we shall use Huygens' principle to show the connection between the eikonal equation and the Hamilton-Jacobi equation. For every point $\mathbf{q}_0$, define the function $S_{\mathbf{q}_0}(\mathbf{q})$ as the optical length of the path from $\mathbf{q}_0$ to $\mathbf{q}$ (see Figure 2). The wave front at time t is given by $\{\mathbf{q} : S_{\mathbf{q}_0}(\mathbf{q}) = t\}$. The vector $\mathbf{p} = \frac{\partial S}{\partial \mathbf{q}}$ is called the *vector of normal slowness of the front.* By Huygens' principle the direction of the ray $\dot{\mathbf{q}}$ is conjugate to the direction of motion of the front, i.e., $\mathbf{p} \cdot \dot{\mathbf{q}} = 1$. Note that these directions do not coincide in an anisotropic medium.

Let us specialize to the case of a monotonically advancing front in an inhomogeneous but isotropic medium (Eq. 1). Here the speed $F(x, y)$ depends only on position (not on direction), and the directions of $\mathbf{p}$ and $\dot{\mathbf{q}}$ coincide.

The action function minimized, $S(\mathbf{q},t)$, is defined as

$$S_{\mathbf{q}_0,t_0}(\mathbf{q},t) = \int_\gamma L dt,$$

along the extremal curve γ connecting the points $(\mathbf{q}_0,t_0)$ and $(\mathbf{q},t)$. Here the Lagrangian

$$L = \frac{1}{F(x,y)}\ ||\partial\gamma/\partial t||$$

is a conformal (infinitesimal) length element, and we have assumed that the extremals emanating from the point $(\mathbf{q}_0,t_0)$ do not intersect elsewhere, i.e., they form a *central field of extremals.* Note that for an isotropic medium the extremals are straight lines, and that for the special case $F(x,y)=1$, the action function becomes Euclidean length.

It can be shown that the vector of normal slowness, $\mathbf{p} = \frac{\partial S}{\partial \mathbf{q}}$, is not arbitrary but satisfies the Hamilton-Jacobi equation

$$\frac{\partial S}{\partial t} = -H\left(\frac{\partial S}{\partial \mathbf{q}}, \mathbf{q}\right), \tag{3}$$

where the Hamiltonian function $H(\mathbf{p},\mathbf{q})$ is the Legendre transformation with respect to $\dot{\mathbf{q}}$ of the Lagrangian function $L(\mathbf{q},\dot{\mathbf{q}})$. Rather than solve the nonlinear Hamilton-Jacobi equation for the action function S (which will give the solution surface $T(x,y)$ to Eq. 2), it is much more convenient to look at the evolution of the phase space $(\mathbf{p},\mathbf{q})$ under the equivalent Hamiltonian system

$$\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}}, \qquad \dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}}.$$

This offers a number of advantages, the most significant being that the equations become linear, and hence trivial to simulate numerically. In the following we shall derive this system of equations for the special case of a front advancing with speed $F(x,y)$ = 1.

4. The Hamilton-Jacobi Skeleton Flow

For the case of a front moving with constant speed, recall that the action function being minimized is Euclidean length, and hence S can be viewed as a Euclidean distance function from the initial curve $\mathcal{C}_0$. Furthermore, the magnitude of its gradient, $||\nabla S||$, is identical to 1 in its smooth regime, which is precisely where the assumption of a central field of extremals is valid.

With $\mathbf{q}=(x,y)$, $\mathbf{p}=(S_x,S_y)$, associate to the evolving plane curve $\mathcal{C}\subset\mathbf{R}^2$ the surface $\tilde{C}\subset\mathbf{R}^4$ given by

$$\tilde{C} := \{(x,y,S_x,S_y) : (x,y)\in C,\ S_x^2+S_y^2=1,\ \mathbf{p}\cdot\dot{\mathbf{q}}=1\}.$$

The Hamiltonian function obtained by applying a Legendre transformation to the Lagrangian $L=||\dot{\mathbf{q}}||$ is given by

$$H = \mathbf{p}\cdot\dot{\mathbf{q}} - L = 1 - (S_x^2+S_y^2)^{\frac{1}{2}}.$$

The associated Hamiltonian system is:

$$\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}} = (0,0), \qquad \dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}} = -(S_x,S_y). \tag{4}$$

$\tilde{C}$ can be evolved under this system of equations, with $\tilde{C}(t)\subset\mathbf{R}^4$ denoting the resulting (contact) surface. The projection of $\tilde{C}(t)$ onto $\mathbf{R}^2$ will then give the parallel evolution of $\mathcal{C}$ at time t, $\mathcal{C}(t)$.

We shall now make use of the fact that all Hamiltonian systems are conservative [16, p. 172]. In particular:

Theorem 1 *The total energy $H(\mathbf{p},\mathbf{q})$ of the Hamiltonian system (4) remains constant along trajectories of (4).*

Proof. The total derivative of $H(\mathbf{p},\mathbf{q})$ along a trajectory $\mathbf{p}(t)$, $\mathbf{q}(t)$ of (4) is given by

$$\frac{dH}{dt} = \frac{\partial H}{\partial \mathbf{p}}.\dot{\mathbf{p}} + \frac{\partial H}{\partial \mathbf{q}}.\dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}}.\frac{\partial H}{\partial \mathbf{q}} - \frac{\partial H}{\partial \mathbf{p}}.\frac{\partial H}{\partial \mathbf{q}} = 0.$$

Thus $H(\mathbf{p},\mathbf{q})$ is constant along any trajectory of (4).

5. Flux and Divergence

The analysis carried out thus far applies under the assumption of a central field of extremals, see Section 3, such that trajectories of the Hamiltonian system do not intersect. Conversely, when shocks form due to the intersection of trajectories, the conservation of energy principle will be violated (energy will be absorbed). As we shall now show, this loss of energy can be used to formulate a robust and very efficient algorithm for shock detection, based on an application of the divergence theorem.

The key is to measure the flux of the vector field $\dot{\mathbf{q}}$, which is analogous to the flow of an incompressible fluid such as water. Note that for a volume with an enclosed surface, an excess of outward or inward flow through the surface indicates the presence of a *source*, or a *sink*, respectively, in the volume. The latter case is the one we are interested in, and we shall use the divergence of the vector field to provide a measure proportional to the net outward flux. More specifically, in physics the divergence of a vector field at a point, $\mathrm{div}(\dot{\mathbf{q}})$, is defined as the net outward flux per unit volume, as the volume about the point shrinks to zero:

$$\mathrm{div}(\dot{\mathbf{q}}) \equiv \lim_{\Delta \mathrm{v}\to 0} \frac{\int_{\mathrm{S}} <\dot{\mathbf{q}},\mathcal{N}>\ \mathrm{ds}}{\Delta \mathrm{v}} \tag{5}$$

Here Δv is the volume, S is its surface and $\mathcal{N}$ is the outward normal at each point on its surface. This definition can be shown to be equivalent to the more common definition of the divergence as the sum of the partial derivatives with respect to each of the vector field's component directions:

$$\mathrm{div}(\dot{\mathbf{q}}) = \frac{\partial \mathrm{q}_{\mathrm{x}_1}}{\partial \mathrm{x}_1} + ... + \frac{\partial \mathrm{q}_{\mathrm{x}_n}}{\partial \mathrm{x}_n} \tag{6}$$

However, Eq. 6 cannot be used at points where the vector field is singular, and hence is not differentiable. These are precisely the points we are interested in, and Eq. 5 offers significant advantages for shock detection. In particular, the numerator, which represents the net outward flux of the vector field through the surface which bounds the volume, is an index computation on the vector field. As we shall see, this is numerically much more stable than the estimation of derivatives in the vicinity of singularities. Further, via the divergence theorem,

$$\int_v \mathrm{div}(\dot{\mathbf{q}})\mathrm{dv} \equiv \int_{\mathrm{S}} < \dot{\mathbf{q}}, \mathcal{N} > \mathrm{ds}. \qquad (7)$$

Hence, the net outward flux through the surface which bounds a finite volume is just the volume integral of the divergence of the vector field within that volume. Locations where the flux is negative, and hence energy is lost, correspond to sink points or shocks.

6. Numerical Simulations

We now apply the above theory to formulate an efficient algorithm for simulating the eikonal equation, while tracking the shocks which form. We shall later discretize the flux computation of Eq. 7 to formulate a very efficient algorithm for shock detection. This can be used to provide an explicit stopping condition for the eikonal evolution.

Recall that since the approach is a Lagrangian one, marker particles will have to first be placed along the initial curve, which in our simulations is assumed to be a simple closed curve in the plane.[1] The evolution of marker particles is then governed by Eq. 4. With $\mathbf{q} = (x, y)$, $\mathbf{p} = (S_x, S_y) = \nabla S$, the system of equations $\{\dot{S}_x = 0, \dot{S}_y = 0; \quad \dot{x} = -S_x, \dot{y} = -S_y\}$ gives a gradient dynamical system. The second equation indicates that the trajectory of the marker particles will be governed by the vector field obtained from the gradient of the Euclidean distance function S, and the first indicates that this vector field does not change with time, and can be computed once at the beginning of the simulation. Projecting this 4D system onto the (x, y) plane for each instance of time t will give the evolved curve $\mathcal{C}(t)$.

In order to obtain accurate results, three numerical issues need to be addressed. First, in order to obtain a dense sequence of marker particles, a continuous representation of the initial shape's boundary ($T(x, y) = 0$, see Figure 1) is needed. Second, it is possible for marker particles to drift apart in rarefaction regions, i.e., concave portions of the curve may fan out. Hence, new marker particles must be interpolated when necessary. Third, whereas finite central differences are adequate for estimating the gradient of the

[1]The method would also extend naturally to open curves, where an outward distance function would have to be defined.

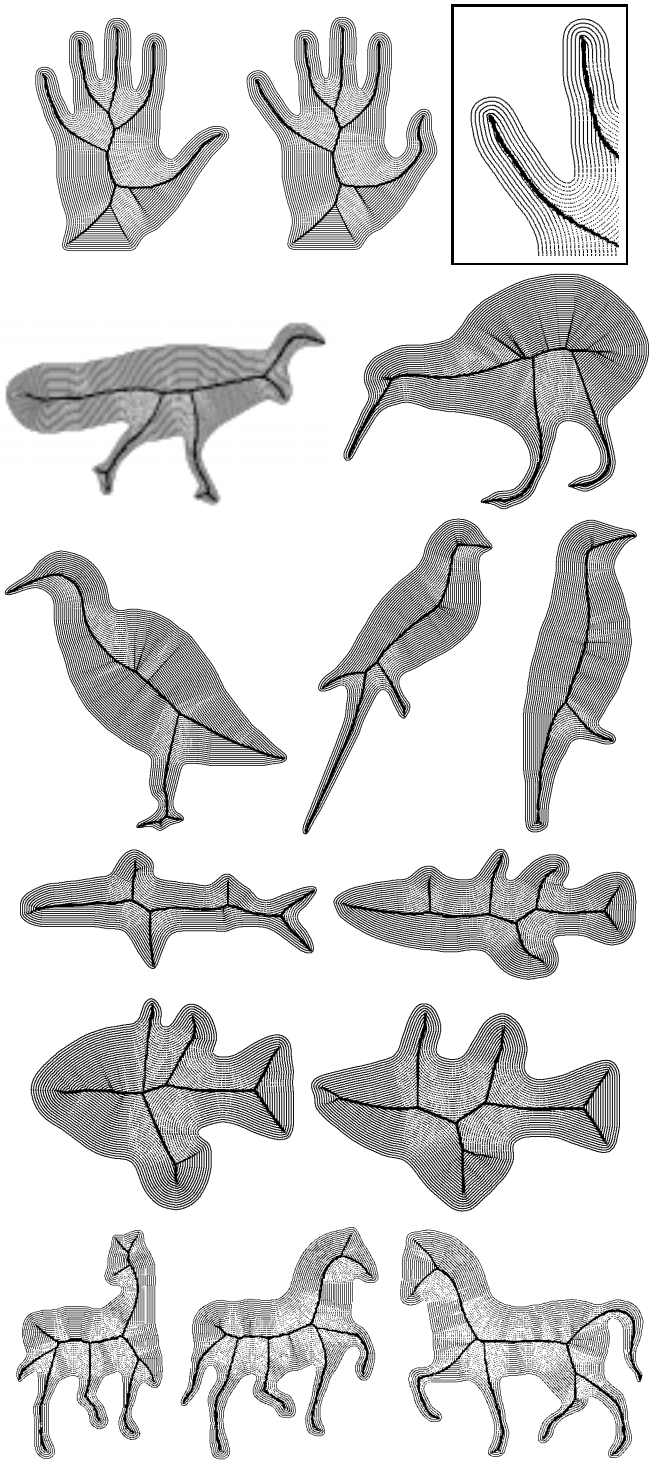

Figure 3: The evolution of marker particles under the Hamiltonian system. The initial particles are placed on the boundary, and iterations of the process are superimposed. These correspond to level sets of the solution surface $T(x, y)$ in Figure 1. Individual marker particles are more clearly visible in the zoom-in on the fingers of the hand (top right).

Euclidean distance function in its smooth regime, such estimates will lead to errors near singularities, where S is not differentiable. Hence, we use ENO interpolants for estimating derivatives [14]; the key idea is to obtain information from the "smooth" side, in the vicinity of a singularity. The algorithm may now be stated as follows:

1. Take as the initial curve $T(x(s), y(s)) = 0$, the given boundary of an object, assumed to be a simple closed curve in the plane.
2. Create an ordered sequence of marker particles at positions Δs apart along the boundary.
3. Compute a Euclidean distance transform, where each grid point in the interior of the boundary is assigned its Euclidean distance to the closest marker particle.
4. For each grid point in the interior of the boundary compute and store the components of the vector field ∇S, using ENO interpolants.

5. Do for **step** from **0** to **TOTALSTEPS** $\{$
  Do for **particle** from **0** to **NPARTICLES** $\{$
    • Update the particle's position based on ∇S at the closest grid point:
      $x(step+1) = x(step) - \Delta t \times S_x$,
      $y(step+1) = y(step) - \Delta t \times S_y$
    • **if** (Distance(particle,next_particle) $> a\Delta s$){
      interpolate a new particle in between.
    }
  }
}

In our experiments, we have used a piecewise circular arc representation of the boundary, obtained using the contour tracer developed in [22], on the signed distance transform of the original binary shape. The distance transform is blurred very slightly to combat discretization. The birth of new marker particles (step 5) is also based on circular arc interpolation. Figure 3 depicts the evolution of marker particles, with speed $F = 1$, for several different shapes. For all simulations, the spacing Δs of initial marker particles is 0.25 pixels, the spacing criterion for interpolating a new particle in the course of the evolution is $a\Delta s$ = 0.75 pixels, and the resolution of the Euclidean distance transform S is the same as that of the original binary image. The timestep Δt is 0.5 pixels, and results for every second iteration are saved. The superposition of all the level curves gives the solution surface $T(x, y)$ in Figure 1. It is important to note that in principle higher order interpolants can be used for the placement of marker particles, and the resolution of the exact distance transform is not limited by that of the original (binary) shape.

The results are comparable to those obtained using higher order ENO implementations, although the algorithm is computationally more efficient (linear in the number of marker particles). Informal timing experiments indicate that the efficiency of the algorithm exceeds that of level set methods, except under the "fast marching" implementation, with which it compares favorably. However, when shock detection is included, the Hamiltonian approach has important conceptual and computational advantages. In particular, in contrast with level set approaches, topological splits are not explicitly handled, but shocks (collisions of marker particles) are. In effect, the marker particles are jittered back and forth along the crest lines of the distance function S, leading to the thick traces in Figure 3.

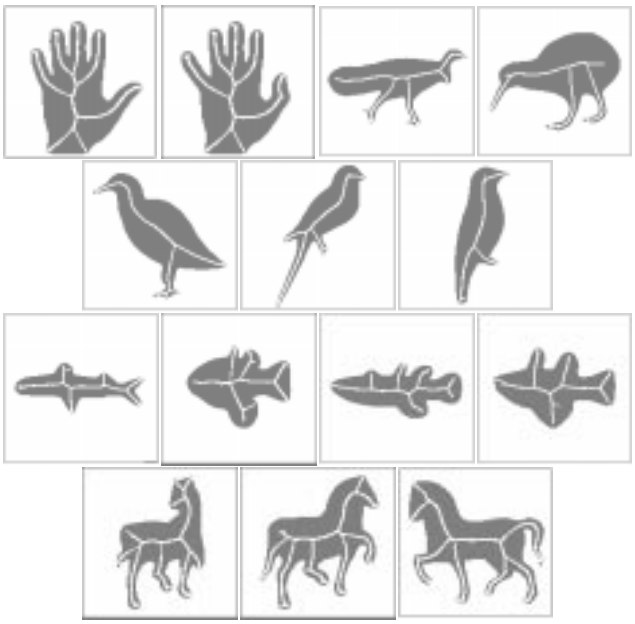

Figure 4: A divergence-based skeleton, superimposed in white on the original binary shapes (shown in grey). Comparing with the Hamiltonian system based flows in Figure 3, these maps can be used to formulate an explicit stopping condition for the individual marker particles.

We now turn to the implementation of the flux computation (Eq. 7) detailed in Section 5. This turns out to be extremely straightforward and efficient to implement, because the computations are local and hence parallelizable. For each grid point consider a small disc in 2D (or a sphere in 3D) centered at the point, such that it passes through its nearest neighboring grid points, i.e., its radius is equal to the grid point spacing Δd. Now, over all neighboring grid points, compute the sum of the inner products of the outward normals to the disc and the vector field $(-S_x, -S_y)$ (in 2D), or the sphere and the vector field $(-S_x, -S_y, -S_z)$ (in 3D). Finally, mark those points where the net outward flux (the volume integral of the divergence) is negative, as sink points or shocks.

Figure 4 illustrates this computation in 2D, using a 3x3 neighborhood, for the same shapes as before. Note that these computations use the same signed distance function

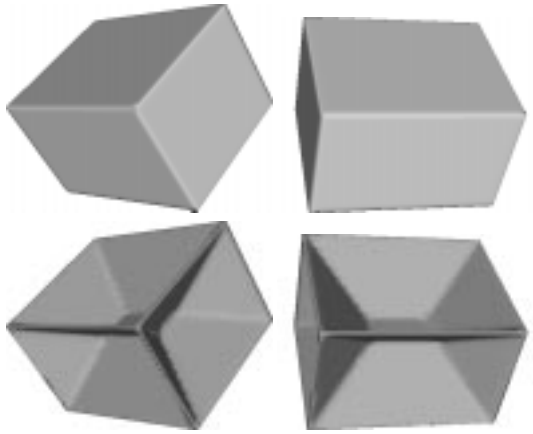

Figure 5: FIRST ROW: Two views of a 3D box. SECOND ROW: The corresponding divergence-based 3D skeletons.

as the earlier marker particle evolutions, but are otherwise entirely independent. Hence, they can be used to provide an explicit stopping condition for the marker particles, should this be desired. Figures 5, 6 and 7 illustrate the divergence-based computation in 3D, using a 3x3x3 neighborhood, on volumetric data of increasing complexity. The box data is synthetic; the brain ventricles and the outer surface of the brain were obtained using surface evolution based segmentation techniques on volumetric MR data. Whereas alternate approaches based on Voronoi techniques provide a topologically organized set of skeletal branches or faces [13, 12], their complexity increases with the number of points on the bounding curve or surface. More significantly, heuristic measures of significance are used for pruning, in order to obtain reasonable results. In contrast, the notion of divergence provides a very natural measure of significance. Furthermore, since the computation is purely local, the 2D or 3D implementations are linear in the number of pixels or voxels in the array; the 3D version takes less than 20 seconds on a 400 MHz Pentium. Whereas naive thresholding can, in principle, yield skeletons which are disconnected or have holes, it is straightforward to extend the method to yield skeletons which are homotopic with the original object. The basic idea is to use the divergence to guide a 2D or 3D thinning process. This extension will be described in future work.

7. Conclusions

This paper makes two main contributions. First, a new algorithm for simulating the eikonal equation has been introduced. The method is rooted in Hamiltonian physics and offers a number of computational advantages when it comes to shock tracking. Second, based on the violation of a conservation of energy principle at singular points of the Hamiltonian system, we have introduced a very efficient and robust algorithm for shock detection. Here the key idea is to measure the net outward flux of a vector field per unit volume, and to detect locations where energy is lost.

In future work we plan to further develop the eikonal equation simulation in 3D, as well as the divergence implementation. The current implementation of divergence is purely local, and involves fixed size discs, which suffices to demonstrate a proof of concept. In closing, we note that in related recent work, a wave propagation framework on a discrete grid has been proposed for curve evolution and mathematical morphology [24], and that vector fields rooted in magneto-statics have also been used for extracting symmetry and edge lines in greyscale images [7].

Figure 6: FIRST ROW: Two views of the ventricles of a brain, obtained using surface evolution based segmentation techniques on volumetric MR data. SECOND ROW: The corresponding divergence-based 3D skeletons.

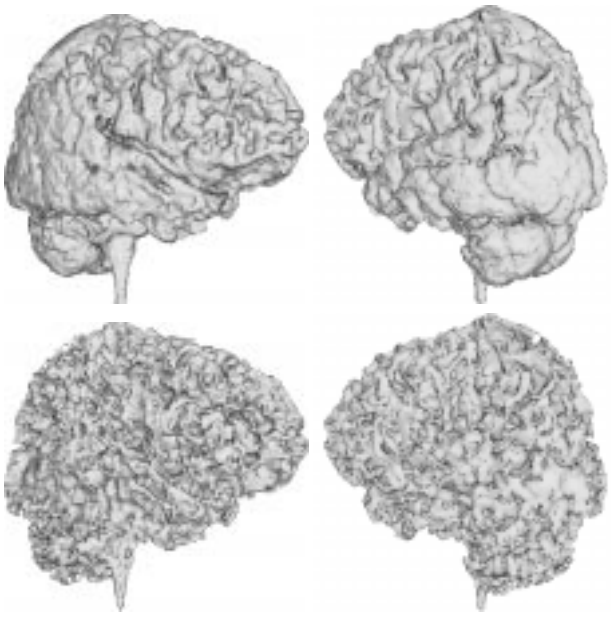

Figure 7: FIRST ROW: Two views of the outer surface of a brain, obtained using surface evolution based segmentation techniques on volumetric MR data. SECOND ROW: The corresponding divergence-based 3D skeletons.

Acknowledgements This work was supported by NSERC, FCAR, NSF, AFOSR, ARO, and a Multi University Research Initiative grant.

References

[1] V. Arnold. *Mathematical Methods of Classical Mechanics, Second Edition*. Springer-Verlag, 1989.

[2] H. Blum. Biological shape and visual science. *J. Theor. Biol.*, 38:205–287, 1973.

[3] R. Brockett and P. Maragos. Evolution equations for continuous-scale morphology. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, San Francisco, CA, March 1992.

[4] A. R. Bruss. The eikonal equation: Some results applicable to computer vision. In B. K. P. Horn and M. J. Brooks, editors, *Shape From Shading*, pages 69–87, Cambridge, MA, 1989. MIT Press.

[5] V. Caselles, J.-M. Morel, G. Sapiro, and A. Tannenbaum, editors. *IEEE Transactions on Image Processing, Special Issue on PDEs and and Geometry-Driven Diffusion in Image Processing and Analysis*, 1998.

[6] M. G. Crandall, H. Ishii, and P.-L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27(1):1–67, 1992.

[7] A. D. J. Cross and E. R. Hancock. Scale-space vector fields for feature analysis. In *Conference on Computer Vision and Pattern Recognition*, pages 738–743, June 1997.

[8] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *Fifth European Conference on Computer Vision*, volume 1, pages 379–393, 1998.

[9] B. B. Kimia, A. Tannenbaum, and S. W. Zucker. Shape, shocks, and deformations I: The components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224, 1995.

[10] R. Kimmel, K. Siddiqi, B. B. Kimia, and A. Bruckstein. Shape from shading: Level set propagation and viscosity solutions. *International Journal of Computer Vision*, 16(2):107–133, 1995.

[11] C. Lanczos. *The Variational Principles of Mechanics*. Dover, 1986.

[12] M. Näf, O. Kübler, R. Kikinis, M. E. Shenton, and G. Székely. Characterization and recognition of 3d organ shape in medical image analysis using skeletonization. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, 1996.

[13] R. L. Ogniewicz and O. Kübler. Hierarchic voronoi skeletons. *Pattern Recognition*, 28:343–359, 1995.

[14] S. Osher and C.-W. Shu. High-order essentially non-oscillatory schemes for Hamilton-Jacobi equations. *SIAM Journal of Numerical Analysis*, 28:907–922, 1991.

[15] S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[16] L. Perko. *Differential Equations and Dynamical Systems*. Springer-Verlag, 1986.

[17] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM. J. Numer. Analy.*, 29(3):867–884, June 1992.

[18] G. Sapiro, B. B. Kimia, R. Kimmel, D. Shaked, and A. Bruckstein. Implementing continuous-scale morphology. *Pattern Recognition*, 26(9), 1992.

[19] J. Sethian. *Level Set Methods: evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, Cambridge, 1996.

[20] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci. USA*, 93:1591–1595, February 1996.

[21] J. Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Conference on Computer Vision and Pattern Recognition*, pages 136–142, June 1996.

[22] K. Siddiqi, B. B. Kimia, and C. Shu. Geometric shock-capturing eno schemes for subpixel interpolation, computation and curve evolution. *Graphical Models and Image Processing*, 59(5):278–301, September 1997.

[23] Z. S. G. Tari, J. Shah, and H. Pien. Extraction of shape skeletons from grayscale images. *Computer Vision and Image Understanding*, 66:133–146, May 1997.

[24] H. Tek and B. B. Kimia. Curve evolution, wave propagation and mathematical morphology. In *Fourth International Symposium on Mathematical Morphology*, June 1998.

[25] R. van den Boomgaard and A. Smeulders. The morphological structure of images: The differential equations of morphological scale-space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11):1101–1113, November 1994.